Graph-based Poisson learning for image co-segmentation

Faculty Mentor: Professor Jeff Calder
Student researcher: Siqi Ke

1. Introduction

Image segmentation is an essential process in computer vision using in video surveillance, medical imaging, and self-driving cars, to name a few applications. Image segmentation separates the main parts of a digital image, including one or more objects of interest, from the background of the image, which helps to recognize objects by only determining the properties of the specific segments. Currently, successful methods for image segmentation include, for example, level-set approaches, graph-cut based spectral approaches, and, more recently, approaches based on deep learning.

Within many different methods for image segmentation, a process called *co-segmentation* has been widely used in medical image segmentation (see, e.g., [6]). Co-segmentation uses the *semi-supervised* technique, requiring the users to label some points for the computer. By doing this, the algorithm receives the prompt of how to cluster the data. After learning from the user-provided data, the segmentation algorithm will usually perform better than unsupervised segmentation algorithms, in which improves the accuracy of image co-segmentation.

Graph-based semi-supervised learning algorithms take the task of image co-segmentation. Graph-based learning (in particular, spectral clustering) was used for unsupervised image segmentation in [7], and a graph-based Ginzburg-Landau approach for co-segmentation was proposed in [3]. One challenge with using graph-based methods for image co-segmentation is that the user may provide very little information within each region, so we are left with a semi-supervised learning problem with very few labels, which has been a very active field of research over the last decade. The widely used Laplacian learning [10], which finds the harmonic extension of the labels on the graph, performs very poorly at low label rates [5], and many new methods have been proposed for low label-rate problems, including higher-order Laplacians [9], p-Laplacian methods [2], reweighted

Laplacians [8], the centered-kernel method [4], and most recently Poisson learning [1]. The results in the recent paper [1] show that Poisson learning performs better than all other methods for graph based semi-supervised learning at low label rates. The idea behind Poisson learning is that the assignment of label values in Laplacian learning is replaced by the placement of sources and sinks in a graph Poisson equation. The method is more stable and informative at low label rates, and it was shown in [1] that the method gives state-of-the-art results on MNIST, FashionMNIST, and Cifar-10 for image classification with very few labeled examples (down to 1 label per class). Poisson learning is also very simple to formulate and efficient to solve, as the equation is a linear positive semi-definite system of equations. It is thus very natural to consider applying Poisson learning to the problem of image co-segmentation.

In this paper, we aim to proof that Poisson learning outperforms all other methods when applied to the problem of image co-segmentation. We hypothesize that the accuracy score of Poisson learning will be higher than other algorithms, such as Laplace learning.

2. Poisson Learning in image co-segmentation: Poisson segmentation

Applying Poisson learning to do the image segmentation consists of the following steps:

(1) Generate a weighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ for the given image. The weight of each edge is set up based on the similarity between the connected notes, which follows the techniques states in [7]

(2) Choose points from the image as a given message to the algorithm. Transform the location of the points into indexes that correspond to the weighted graph and create a new array to label the points.

(3) Apply the Poisson learning. A weight matrix $\mathbf{W}$, an index list of chosen points $\mathbf{I}$, and a corresponding label list $\mathbf{C}$ are required to be provided as the inputs. The algorithm returns a list of returned labels.

(4) Make the segmentation mask for the given image with the returned labels.

More details of the image segmentation process are shown in the coming subsection.

2.1. **The weighting function.** To apply graph-based learning to image co-segmentation, we have to construct a graph over the normalized image. We plan to follow the techniques in [7], which use the weights

$$w_{ij} = \begin{cases} e^{-\frac{\|I_i - I_j\|_2^2}{\sigma_I^2} - \frac{\|X_i - X_j\|_2^2}{\sigma_X^2}}, & \text{if } \|X_i - X_j\| \le r \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where I_i denotes the intensity, and X_i the spatial coordinates, of pixel i in the image. The parameters σ_I, σ_X and r are user-tuned and control the sparsity and connectivity structure in the resulting graph. Precisely, σ_I controls the similarity of intensity, color and texture between nodes while σ_X and r are spatially related. Experiments about how those parameters affect the segmentation results are shown in the experimental results section. The idea is to connect nearby pixels in the image by an edge whose weight depends on the similarity in the intensity values of the image. It is also possible to use more sophisticated weights based on similarity of local image patches, as was done in [3].

2.2. **Poisson learning for segmentation.** After constructing the weight matrix as in (1), denoted $W = (w_{ij})_{i,j=1}^n$, were n is the number of pixels in the image, we plan to apply Poisson learning [1] to the weight matrix $\mathbf{W}$ for image co-segmentation. For each image, we will decide on the number of objects of interest to segment and will provide an image mask with several pixels identified within each object. These will be provided as labeled data for Poisson learning in the form of a list of indexes $\mathbf{I}$ and a list of labels $\mathbf{C}$, which will be run to give the final co-segmentation. The Poisson learning equation has the form $Lu = f$, where L is the graph Laplacian matrix, f is a source term that includes all the information about the labeled image pixels, and the solution u gives the final co-segmentation. We will use the Python package GraphLearning `https://github.com/jwcalder/GraphLearning`, developed by Dr. Calder, which includes subroutines for solving the Poisson learning problem.

3. Experimental Results

3.1. **Tolerance of Poisson learning to the weight matrix.** In Normalized Cuts and Image Segmentation [7], the proper way to assign weights for edges is discussed. Since images are very and the weight of edges is decided by the feature similarity, which is hard to quantify, the best weighting function is possibly different from region to region in the same image. Therefore, it is not easy to find the most appropriate assignment function for each graph in practical applications, which requires the graph-learning algorithm to be more tolerant to the weight matrix.

We tested the tolerance of Poisson learning for the weight matrix on the image "Cameraman." The original Cameraman is a 512×512 image with three channels. The objects of interest are the cameraman and the background. A click-pick tool is built to simplify choosing points and make the given message more random. 2 points are given per class. To reduce time costs, we do the parameter testing on a smaller image that we make a sub-sample of the Cameraman. In order to clearly show the effect of each variable on the segmentation results, we designed a total of three sets of control experiments. In each set of experiments, we kept two variables constant and varied the remaining one variable separately. The choice of the variables is in 0.005, 0.01, 0.1 for σ_I, 2, 3, 5 for σ_X, and 3, 5, 10 for r. The above testing values are chosen from the normalized cut paper [7]. The segmentation results are shown in figure 1. The runtime for Poisson learning is also recorded in table 1. (The time is only for the Poisson learning process, which does not include the time for picking points and building the weighted graph.)

Term	Runtime	Term	Runtime	Term	Runtime
(0.005, 3, 5)	41.4665s	(0.01, 2, 5)	36.0868s	(0.01, 3, 3)	16.6142s
(0.01, 3, 5)	38.3557s	(0.01, 3, 5)	38.3557s	(0.01, 3, 5)	38.3557s
(0.1, 3, 5)	35.5601s	(0.01, 5, 5)	40.4677s	(0.01, 3, 10)	131.6153s

TABLE 1. Runtimes of Poisson learning process for terms $(\sigma_I,\ \sigma_X,\ r)$

According to the weighting function, decreasing σ_I and σ_X will amplify the variability of weights caused by the difference between notes. Furthermore, increasing r allows more relations between pairs of points to be evaluated. Knowing these characteristics, we analyze the graph segmentation results obtained in the parameter test. Except for the (0.1, 3, 5) term that blurred edges distinction due to the large σ_I, Poisson learning performs well in the rest segmentation. This result demonstrates that Poisson learning can deal well with the errors introduced by the weight function. Moreover, in term (0.01, 2, 5), (0.01, 3, 3), and (0.01, 3, 10), the results show that they

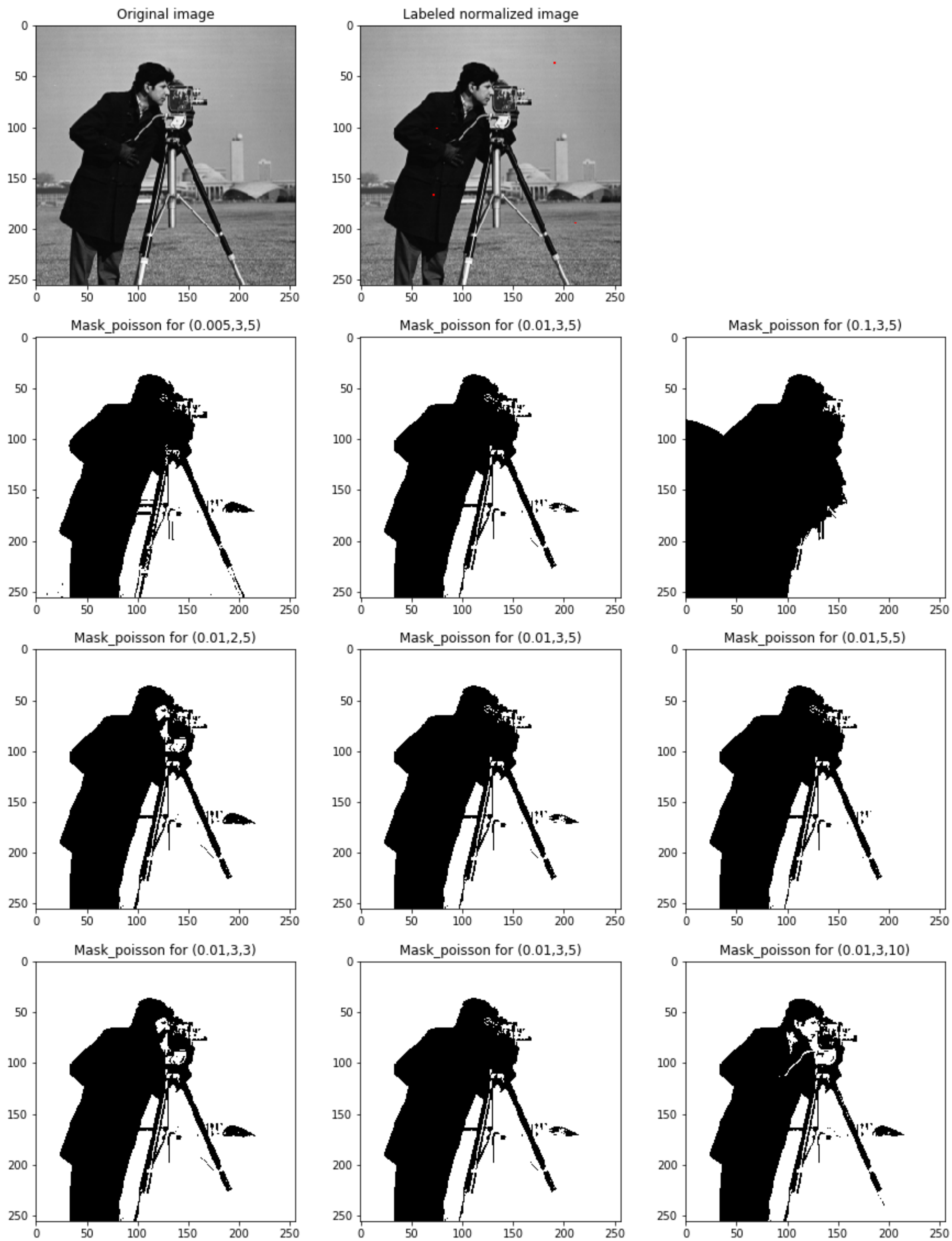

FIGURE 1. Parameter testing for the weighting function on a 256×256 Cameraman image with 3 channels. $(\sigma_I,\ \sigma_X,\ r)$

have included the photographer's face in the background, which reminds that over-reinforcing distinctions between points, or evaluating too many, too few point combinations, will lead to over-segmentation. As for the runtime issue, the only noticeable change of the runtime is raised by the r parameter, which is under expectation since r is the only parameter that affects the number of valid entries in the weight matrix $\mathbf{W}$.

3.2. **Poisson learning V.S. Laplace learning.** In Poisson Learning: Graph-Based Semi-Supervised Learning At Very Low Label Rates [1], the authors mention that Poisson learning mainly compensates for the poor performance of the traditional Laplacian learning when only a minimal amount of information is provided. Therefore, we designed the experiments to verify whether Poisson learning also shows superiority in image segmentation when only a few labels are available.

We tested the performance of the Poisson algorithm and Laplace algorithm on two images with only a few labeled points, respectively. The first photo is the black and white 256×256 image Cameraman used to test the tolerance of Poisson's algorithm above, and the other is a color 150×150 image the Beach. We want two classes in the Cameraman: the cameraman and the background, and four classes in the Beach: the sky, the sea, the beach, and the person. $\sigma_I = 0.01,\ \sigma_X = 3,\ r = 5$ are used to build the weighted graph. The same message (same points with same labels) is given to the two algorithms: 1, 2, and 5 labeled points per class every time. The segmentation results are shown in Figure 2-7. The runtimes for Poisson learning and Laplacian learning are also recorded in Table 2. (The time is only for the Poisson learning or Laplacian learning process, which does not include the time for picking points and building the weighted graph.)

Comparing the segmentation results of two groups of images in different situations and with different algorithms, it is easy to see the advantages of Poisson's algorithm. When only one labeled point is provided (Figure 2), the Poisson algorithm delineates a smoother photographer profile. The Laplacian algorithm misclassifies some scattered pixel points to varying degrees. When there are only 1 and 2 labels provided (Figure 3&5), the boulder where the visitor is sitting is divided into the human part. In the Cameraman (Figure 2&6), some of the grass parts are also classified as the human part. Looking at Table 2, Poisson learning takes less time compared to Laplacian leaning, which also demonstrated the superiority of Poisson's algorithm.

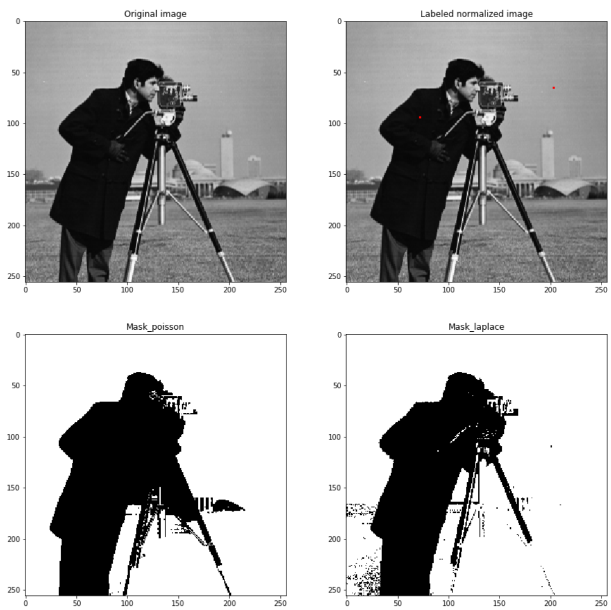

FIGURE 2. 1 label per class for the 256×256 Cameraman

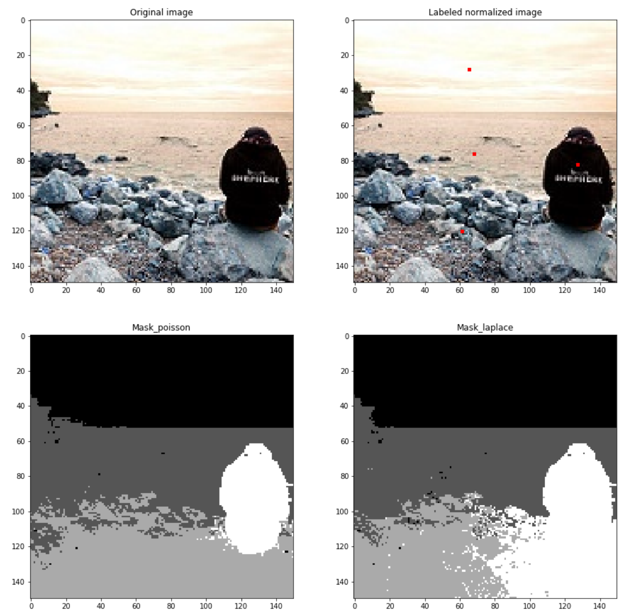

FIGURE 3. 1 label per class for the 150×150 Beach

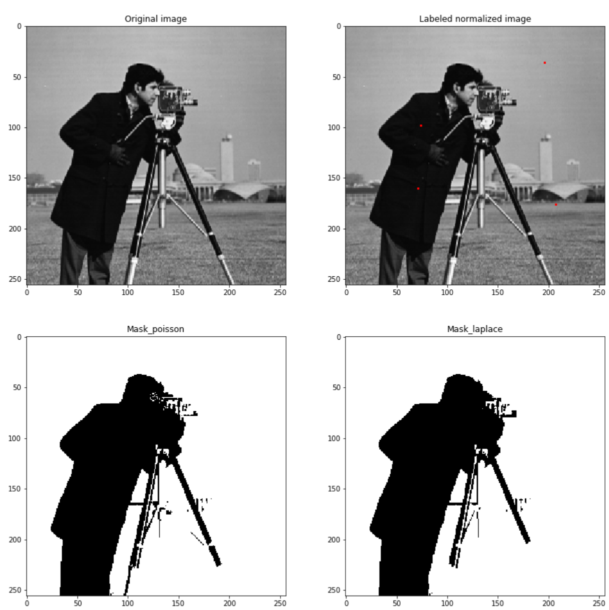

FIGURE 4. 2 labels per class for the 256×256 Cameraman

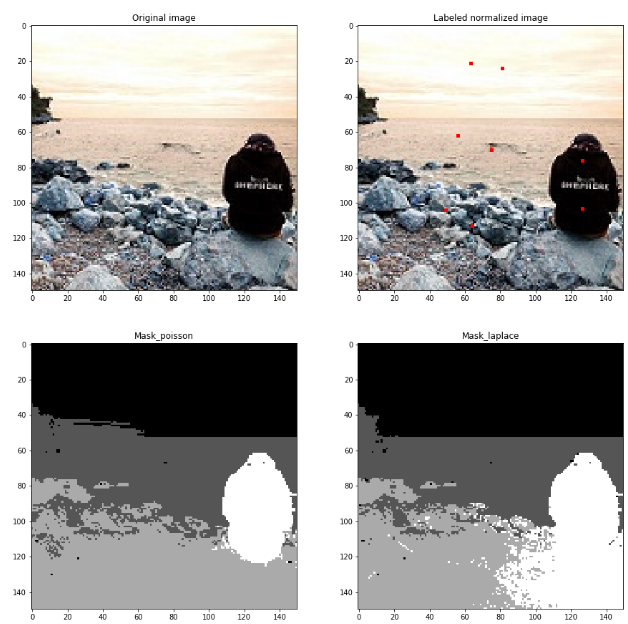

FIGURE 5. 2 labels per class for the 150×150 Beach

FIGURE 6. 5 labels per class for the 256×256 Cameraman

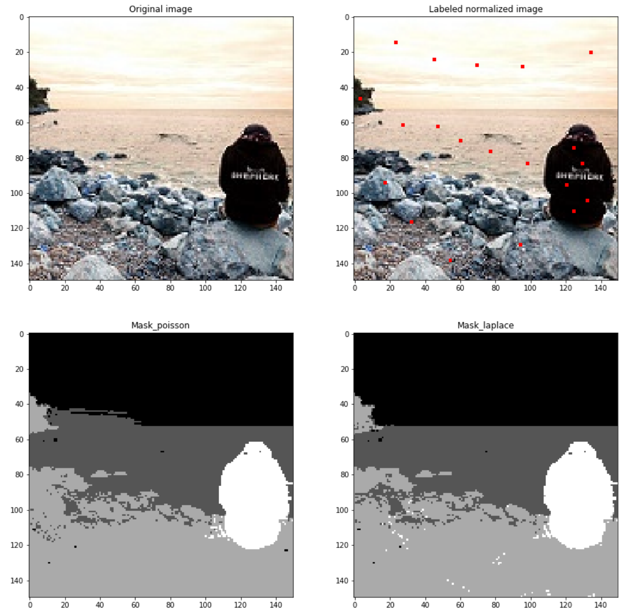

FIGURE 7. 5 labels per class for the 150×150 Beach

Image	Number of given labels	Poisson Runtime	Laplace Runtime
Cameraman	1	30.7218s	326.3801s
Cameraman	2	30.7303s	264.0041s
Cameraman	5	27.6819s	223.9135s
Beach	1	15.0974s	43.9028s
Beach	2	13.0912s	68.0221s
Beach	5	14.5414s	73.2793s

TABLE 2. Runtimes of Poisson learning and Laplacian learning ($\sigma_I = 0.01,\ \sigma_X = 3,\ r = 5$)

4. Conclusion

We found an efficient and accurate method, Poisson segmentation, for image segmentation. The method applies the graph-based semi-supervised Poisson learning on the weighted graph generated by the image and gives a good segmentation based on our requirement. We tested the tolerance of the Poisson learning with multi-weight metrics and the accuracy of the Poisson segmentation at very low label rates showing that the method works well.

REFERENCES

[1] J. Calder, B. Cook, M. Thorpe, and D. Slepčev. Poisson Learning: Graph Based semi-supervised learning at very low label rates. *Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119*, 2020.

[2] A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan. Asymptotic behavior of ℓ_p-based Laplacian regularization in semi-supervised learning. In *Conference on Learning Theory*, pages 879–906, 2016.

[3] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner, and A. G. Percus. Multiclass data segmentation using diffuse interface methods on graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1600–1613, 2014.

[4] X. Mai and R. Couillet. Random matrix-inspired improved semi-supervised learning on graphs. In *International Conference on Machine Learning*, 2018.

[5] B. Nadler, N. Srebro, and X. Zhou. Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data. *Advances in neural information processing systems*, 22:1330–1338, 2009.

[6] D. L. Pham, C. Xu, and J. L. Prince. Current methods in medical image segmentation. *Annual review of biomedical engineering*, 2(1):315–337, 2000.

[7] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[8] Z. Shi, S. Osher, and W. Zhu. Weighted nonlocal Laplacian on interpolation from sparse data. *Journal of Scientific Computing*, 73(2-3):1164–1177, 2017.

[9] X. Zhou and M. Belkin. Semi-supervised learning by higher order regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 892–900, 2011.

[10] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine learning (ICML-03)*, pages 912–919, 2003.